Plan2Fund — Updated User Journey (Business + Data Architecture)

Version: 2025-09-01

1) Purpose

This document aligns the desired business flow with the current Next.js repository. It explains the screens, the data objects they read/write in Supabase, and how payments/exports are gated with feature flags. It replaces any prior draft and serves as the source of truth for the MVP build and polishing agent.

2) Stack Snapshot

• Web: Next.js 14 (pages router present), TypeScript, Tailwind
• Backend: Supabase (Postgres + Auth + Storage), API routes (Server Actions later if app/ router is introduced)
• Payments: Stripe Checkout (feature-flagged)
• Email: Resend (order confirmations; later notifications)
• Hosting: Vercel
• Flags: CHECKOUT_ENABLED, EXPORT_ENABLED, AI_ENABLED

3) Primary Flow (Business View → Data Touchpoints)

A. Landing “Welcome/Choose Path” (/)
– Business: Explain value, let the user choose Find Funding (Reco) or Business Plan.
– Data: Anonymous view; optional session_id cookie.

B. Recommendation Wizard (/reco)
– Business: One question per screen, show progress; ends with ranked Top-5 programs with reason and % fit.
– Data: reco_sessions (one per run), reco_answers (one per question), reco_results (score, reason, program_id).

C. Business Plan Editor (/plan)
– Business: Sidebar chapters, editable sections, autosave, versioning; user can start here directly or via results.
– Data: plan_documents (one per plan), plan_versions (autosave snapshots), uploads (files in storage).

D. Preview & Pricing (/preview, /pricing)
– Business: Completeness meter, complexity score and price; CTA to purchase.
– Data: reads plan_documents + rollup metrics; creates draft order when user proceeds.

E. Checkout (/checkout)
– Business: Stripe Checkout if CHECKOUT_ENABLED=true; else disabled stub.
– Data: orders (pending → paid); payments (provider receipt), webhooks update order statuses.

F. Export (/export)
– Business: Generate DOCX/PDF; if EXPORT_ENABLED=false or unpaid, show masked preview with watermark.
– Data: exports (artifact path, hash), files in Supabase Storage.

G. Thank You & After-Sales (/thank-you)
– Business: Show downloads, email confirmations, prompt for revision request.
– Data: reads last paid order and associated export links.

4) Screens & Components

Global: Header, Footer, LanguageSwitcher, Breadcrumbs (from /reco or /plan onward), ProgressBar, SaveIndicator, Toasts
Reco: WizardStep, QuestionCard, ResultsList, ResultCard
Plan: PlanShell, ChapterSidebar, EditorArea (rich text/markdown), VersionBar, Attachments
Checkout/Export: OrderSummary, PayButton (flagged), ExportLocker, PreviewPane
Utilities: FeatureFlag, RequireAuth (if needed later)

5) Data Model (Supabase)

profiles(id, email, name, created_at)
reco_sessions(id, user_id nullable, started_at, completed_at)
reco_answers(id, session_id, qkey, value)
reco_results(id, session_id, program_id, score, fit_reason, effort_estimate)
plan_documents(id, user_id, title, status, created_at, updated_at)
plan_versions(id, plan_id, content_json, notes, created_at)
uploads(id, user_id, plan_id nullable, path, mime, size, created_at)
orders(id, user_id, plan_id, amount_cents, currency, status[pending|paid|failed], created_at)
payments(id, order_id, provider, provider_id, amount_cents, status, created_at)
exports(id, plan_id, order_id, kind[pdf|docx], path, created_at)
journey_events(id, user_id nullable, name, payload_json, created_at)

6) Feature Flags

CHECKOUT_ENABLED=false → renders PayButton disabled + info tooltip
EXPORT_ENABLED=false → masks exports behind paywall or watermark
AI_ENABLED=false → keeps vector/LLM features off until MVP is live

7) Error States & Non-Technical Explanations

• If Vercel build fails: the app won’t deploy, but your code is safe on GitHub. The agent fixes code, commits again, and Vercel retries automatically.
• If Stripe is off: users can still see prices but cannot pay; that’s intentional until testing is complete.
• Supabase Studio works like a spreadsheet for your database; you can view answers, plans, and orders without coding.

8) Metrics (MVP)

• Wizard completion rate, time to first result
• Plans created, versions per plan, preview-to-checkout CTR
• Paid orders, export success rate

9) Next Steps (for the Agent)

1) Finalize business logic for 3a and 3b (questions catalog, scoring rules, plan chapter schema, autosave cadence) and wire routes/components; document data flow inline.
2) Set up tech stack access (Supabase project + keys, Stripe keys + webhook on /api/stripe/webhook, Resend key). Add .env and feature flags; fix Vercel deployment.
3) Ship a functioning MVP: end-to-end reco→plan→preview→(checkout flagged)→export (flagged). Add minimal tests (smoke + data writes).

10) Agent Execution Prompt (Polish & Build)

Role: Senior Next.js + Supabase + Stripe engineer. Objectives: clean repo, fix build, implement missing pages/components, wire Supabase tables, add API routes, integrate Stripe Checkout (flagged), produce a running MVP on Vercel.

Constraints: keep dependencies minimal, no ORM, use Supabase JS client, use Tailwind, maintain feature flags.

Deliverables: green Vercel deploy, updated README with env setup, SQL for tables, test plan, and checklists mapping to this document.